

Spring Boot实际应用讲解（五）：AOP之请求日志



ZYRzyr (/u/f8ff63b17fc7) + 关注

2017.11.22 16:53* 字数 1141 阅读 312 评论 0 喜欢 2

(/u/f8ff63b17fc7)

文/ZYRzyr (<https://www.jianshu.com/u/f8ff63b17fc7>)
原文链接:<http://www.jianshu.com/p/93216bf41182>
(<https://www.jianshu.com/p/93216bf41182>)

本文提纲

- 一、AOP超简介绍
- 二、实例
- 三、最后

本文运行环境

Ubuntu 16.04 LTS
JDK 8 +
IntelliJ IDEA ULTIMATE 2017.2
Maven 3.5.0
Spring Boot 1.5.8.RELEASE

一、AOP超简介绍

首先强调，AOP 并不仅仅是在 Java 中才有的。

网上有很多对 AOP 的详细介绍，也有很多讲的很好的，这里就不再赘述。一句话描述就是：每当调用某些方法的过程中，都想做同样的事 A，用一种**尽量不重复代码**的方式把 A 插入到这些方法中，这就是 AOP，其中被插入的方法的某个过程（方法执行前、执行后、返回值时、抛出异常时等）即为切点。

AOP 最大的好处就在于：不用修改已有的代码，即每个想做 A 的方法都单独把 A 写进去，同时减少了代码的重复，降低了与业务逻辑无关的代码的耦合。

二、实例

本文就在控制台输出每个客户端请求的详细信息为例展示 AOP 的便利之处。

2.1 添加依赖

要使用 AOP，需要在 pom.xml 中添加如下依赖：

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-aop</artifactId>  
</dependency>
```



2.2 横切类

新建类 HttpAspect.java :

```
package com.zyr.demo.aspect;

import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.AfterReturning;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.aspectj.lang.annotation.Pointcut;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Component;
import org.springframework.web.context.request.RequestContextHolder;
import org.springframework.web.context.request.ServletRequestAttributes;

import javax.servlet.http.HttpServletRequest;

@Aspect
@Component
public class HttpAspect {

    private static final Logger logger = LoggerFactory.getLogger(HttpAspect.class);

    @Pointcut("execution(public * com.zyr.demo.controller..*.*(..))")
    public void logRequest() {
        //no-op
    }

    @Before("logRequest()")
    public void doBefore(JoinPoint joinPoint) {
        ServletRequestAttributes attributes = (ServletRequestAttributes) RequestContextHolder.getRequestAttributes();
        HttpServletRequest request = attributes.getRequest();

        logger.info("-----Request Log Begin-----");
        logger.info("url={}", request.getRequestURL());
        logger.info("method={}", request.getMethod());
        logger.info("ip={}", request.getRemoteAddr());
        logger.info("class_method={}", joinPoint.getSignature().getDeclaringTypeName());
        logger.info("args={}", joinPoint.getArgs());
        logger.info("-----Request Log End-----");
    }

    @AfterReturning(returning = "object", pointcut = "logRequest()")
    public void doAfterReturning(Object object) {
        if (object != null) {
            logger.info("response={}", object.toString());
        }
    }
}
```

1. @Aspect : 表示该类是一个横切类;
2. @Component : 表示将该类放入 Spring 容器中;
3. @Pointcut : 表示切入的地方, 括号内 execution 的括号中表示 com.zyr.demo.controller 包中的所有 public 方法;
4. logRequest() : 由 @Pointcut 表示该方法即为上文提到的事 A, 方法名自定义, 方法体内为空, 表示该方法单纯指定切入点, 具体内容在引用它的地方写, 如下面的 doBefore 和 doAfterReturning ;
5. @Before : 括号中引用了上面的 logRequest(), 表示该方法的切入点与它相同, @Before 表示在被切入的方法开始执行前, 需要先执行该方法体内的代码, 此处即为打印请求信息, 方法名自定义;



1/7/2018

Spring Boot实际应用讲解（五）：AOP之请求日志 - 简书

6. @AfterReturning：括号中的 object 必须与方法的参数名相同，表示被切入的方法的返回值。pointcut 即同上，表示切入点与 logRequest() 的切入点相同，@AfterReturning 表示在被切入的方法返回值时，执行该方法中的内容；

7. 除了@Before 和 @AfterReturning，还有 @After、AfterThrowing、@Around 等切入方式。

(/i
ut

2.3 测试

运行项目，打开 REST Client，测试上一节中的 localhost:8080/user/register，提交成功后，可以看到控制台输出以下信息，说明切入成功，相应的可以测试其它方法：

```
2017-11-22 16:46:23.591 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : -----Request Log Begin-----
-----
2017-11-22 16:46:23.591 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : url=http://localhost:8080/user/register
(https://link.jianshu.com?t=http://localhost:8080/user/register)
2017-11-22 16:46:23.592 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : method=POST
2017-11-22 16:46:23.592 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : ip=127.0.0.1
2017-11-22 16:46:23.592 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect :
class_method=com.zyr.demo.controller.UserController.register
2017-11-22 16:46:23.592 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : args=User{name='bob', age=22,
password='123456'}
2017-11-22 16:46:23.592 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : -----Request Log End-----
-----
2017-11-22 16:46:23.595 INFO 10526 --- [nio-8080-exec-1]
com.zyr.demo.aspect.HttpAspect : response=<200 OK,success,{}>
```

三、最后

本文介绍了 Spring Boot 中使用 AOP 的一个简单例子，可以看见，在 Spring Boot 中很轻松就能使用 AOP。

本文代码已上传至我的GitHub仓库 (<https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo>)，进入以后将branches (<https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo/branches>)切换为5-AOP (<https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo/tree/5-AOP>)即可看见。

- 前篇：
- Spring Boot实际应用讲解（一）：Hello World (<https://www.jianshu.com/p/60f7e025c680>)
 - Spring Boot实际应用讲解（二）：配置详解 (<https://www.jianshu.com/p/d4c7f33c9b37>)
 - Spring Boot实际应用讲解（三）：表单验证 (<https://www.jianshu.com/p/a2b4e61b5532>)
 - Spring Boot实际应用讲解（四）：RESTful API (<https://www.jianshu.com/p/e907595e9d1d>)

文中若有错之处，还请各位批评指正，谢谢！

原作者/ZYRzyr (https://www.jianshu.com/u/f8ff63b17fc7)
原文链接:http://www.jianshu.com/p/93216bf41182
(https://www.jianshu.com/p/93216bf41182)

(https://link.jianshu.com?t=https://101709080007647.bqy.mobi)



(https://link.jianshu.com?t=https://101709080007647.bqy.mobi)

Spring Boot (/nb/18796030) 举报文章 © 著作权归作者所有



ZYRzyr (/u/f8ff63b17fc7) ♂
写了 17755 字，被 32 人关注，获得了 92 个喜欢
(/u/f8ff63b17fc7)

+ 关注

程序猿一枚 技能树加点情况： 移动端—Android—5/5 移动端—iOS—1/5 前端—JavaScript、HTML—...


喜欢的老铁，来一波关注666

赞赏支持

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 2

更多分享

(http://cwb.assets.jianshu.io/notes/images/20055015/



下载简书 App ▶
随时随地发现和创作内容



(/apps/download?utm_source=nbc)



登录 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-comment-form) 发表评论

评论

